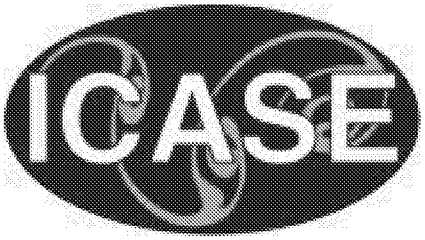
NASA/CR-2001-211230
ICASE Report No. 2001-29



The Solution of Radiation Transport Equations with Adaptive Finite Elements

Linda Stals
Old Dominion University, Norfolk, Virginia

October 2001

The NASA STI Program Office . . . in Profile

Since its founding, NASA has been dedicated to the advancement of aeronautics and space science. The NASA Scientific and Technical Information (STI) Program Office plays a key part in helping NASA maintain this important role.

The NASA STI Program Office is operated by Langley Research Center, the lead center for NASA's scientific and technical information. The NASA STI Program Office provides access to the NASA STI Database, the largest collection of aeronautical and space science STI in the world. The Program Office is also NASA's institutional mechanism for disseminating the results of its research and development activities. These results are published by NASA in the NASA STI Report Series, which includes the following report types:

- TECHNICAL PUBLICATION. Reports of completed research or a major significant phase of research that present the results of NASA programs and include extensive data or theoretical analysis. Includes compilations of significant scientific and technical data and information deemed to be of continuing reference value. NASA's counterpart of peer-reviewed formal professional papers, but having less stringent limitations on manuscript length and extent of graphic presentations.

- TECHNICAL MEMORANDUM. Scientific and technical findings that are preliminary or of specialized interest, e.g., quick release reports, working papers, and bibliographies that contain minimal annotation. Does not contain extensive analysis.

- CONTRACTOR REPORT. Scientific and technical findings by NASA-sponsored contractors and grantees.

- CONFERENCE PUBLICATIONS. Collected papers from scientific and technical conferences, symposia, seminars, or other meetings sponsored or cosponsored by NASA.

- SPECIAL PUBLICATION. Scientific, technical, or historical information from NASA programs, projects, and missions, often concerned with subjects having substantial public interest.

- TECHNICAL TRANSLATION. English-language translations of foreign scientific and technical material pertinent to NASA's mission.

Specialized services that complement the STI Program Office's diverse offerings include creating custom thesauri, building customized data bases, organizing and publishing research results . . . even providing videos.

For more information about the NASA STI Program Office, see the following:

- Access the NASA STI Program Home Page at ***http://www.sti.nasa.gov***

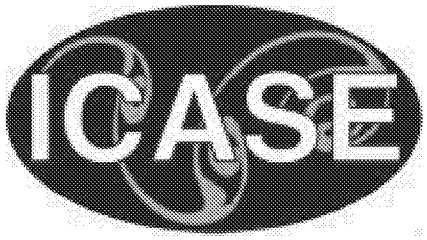
- Email your question via the Internet to help@sti.nasa.gov

- Fax your question to the NASA STI Help Desk at (301) 621-0134

- Telephone the NASA STI Help Desk at (301) 621-0390

- Write to:
  NASA STI Help Desk
  NASA Center for AeroSpace Information
  7121 Standard Drive
  Hanover, MD 21076-1320

NASA/CR-2001-211230
ICASE Report No. 2001-29



The Solution of Radiation Transport Equations with Adaptive Finite Elements

Linda Stals
Old Dominion University, Norfolk, Virginia

ICASE
NASA Langley Research Center
Hampton, Virginia

Operated by Universities Space Research Association

National Aeronautics and
Space Administration

Langley Research Center
Hampton, Virginia 23681-2199

Prepared for Langley Research Center
under Contract NAS1-97046

October 2001

Available from the following:

NASA Center for AeroSpace Information (CASI)
7121 Standard Drive
Hanover, MD 21076-1320
(301) 621-0390

National Technical Information Service (NTIS)
5285 Port Royal Road
Springfield, VA 22161-2171
(703) 487-4650

THE SOLUTION OF RADIATION TRANSPORT EQUATIONS WITH ADAPTIVE FINITE ELEMENTS*

LINDA STALS†

Abstract. We compare the performance of an inexact Newton-multigrid method and Full Approximation Storage multigrid when solving radiation transport equations. We also present an adaptive refinement algorithm and explore its impact on the solution of such equations.

Key words. FAS, multigrid, Newton method, radiation transport

Subject classification. Computer Science

1. Introduction. Interest in the solution of radiation transport equations stems from the modeling of applications found in, for example, combustion, astrophysics and laser fusion. However, features such as strong nonlinearities and large jumps in the coefficients makes these equations difficult to solve and they can consume a lot of computational resources if efficient solution techniques are not used. Two examples of efficient solution methods are the inexact Newton-multigrid method and the Full Approximation Storage multigrid (FAS); both of which are nonlinear multigrid techniques.

The solution of radiation transport equations usually has a wave front. Adaptively refined grids are well suited to capture the information along the front and thus give high resolution results.

In this paper we compare the performance of an inexact Newton-multigrid method and the FAS method for the solution of time-dependent radiation transport equations. We also explore the use of adaptive refinement techniques.

2. Physical Model. Under certain assumptions, such as isotropic radiation, optically thick material and temperature equilibrium, radiation transport may be modeled by the equation:

$$\frac{\partial E}{\partial t} - \nabla.(D(E)\nabla E) = 0 \qquad \text{on } \Omega \times I, \tag{2.1}$$

where E is the radiation energy.

More physically meaningful models of radiation transport are represented by systems of equations like those described in [4, 12, 14]; however, Equation (2.1) contains many of the features seen in the more general system, such as strong nonlinearities and large jumps in the coefficients, and therefore is a good place to start our investigation into different solution techniques.

One definition of the diffusivity term, $D(E)$, is:

$$D(E) = D_1(E) = Z^{\alpha} E^{\beta}, \tag{2.2}$$

with $\alpha < 0$, $0 \leq \beta \leq 1$. Typically α is taken to be -1 or -3 while β is taken to be $1/4$ or $3/4$. Z is the atomic mass number and may vary within the domain to reflect inhomogeneities in the material. The constant β controls the strength of the nonlinearity while α affects the size of the jumps in the coefficients.

*This research was supported by the U.S. Department of Energy under the ASCI ASAP program (subcontract B347882 from the Lawrence Livermore National Laboratory) and by the National Aeronautics and Space Administration under NASA Contract No. NAS1-97046 while the author was in residence at ICASE, NASA Langley Research Center, Hampton, Virginia 23681-2199, USA.

†Department of Computer Science, Old Dominion University, Norfolk, VA 23529-0162, USA and ICASE, NASA Langley Res. Ctr, Hampton, VA 23681-2199, USA.

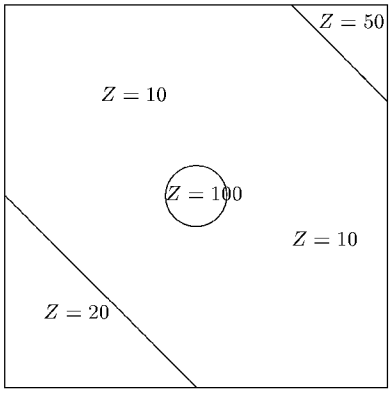


FIG. 2.1. *The values for the atomic mass number Z depend on the topology of the material. In our model problem, we define Z as shown above.*

The definition of $D(E)$ given in Equation (2.2) may produce results that show the energy moving through the system at a rate faster than the speed of light. Consequently a flux limiter is included to slow down the movement and the diffusivity term is rewritten as:

$$D(E) = D_2(E) = \left(\frac{1}{Z^\alpha E^\beta} + \frac{|\nabla E|}{E} \right)^{-1}. \tag{2.3}$$

The domain, Ω, is a square domain ($[0,1] \times [0,1]$) with the following mixture of Newton and Neumann boundary conditions;

$$\begin{aligned} &\partial E/\partial n = 0 && \text{on } ,_N \times I, \\ &n^T D(E) \nabla E + E/2 = 2 && \text{on } ,_{F_0} \times I, \\ &n^T D(E) \nabla E + E/2 = 0 && \text{on } ,_{F_1} \times I, \end{aligned}$$

where $,_N$ represents the lines $y = 0$ and $y = 1$, $,_{F_0}$ is the line $x = 0$ and $,_{F_1}$ is the line $x = 1$, n is the outward unit normal and I is the time interval.

In our model problem we take Z to be 10.0 except in the following regions: $\sqrt{(x-0.5)^2 + (y-0.5)^2} \leq 0.125$, $y \leq 0.5 - x$ and $y \geq 1.75 - x$ where $Z = 100$, $Z = 20$ and $Z = 50$ respectively. See Figure 2.1.

Additional papers that look at these equations include [3, 4, 5, 7, 12, 16] and their accompanying references. The radiation transport model described here is very similar to that presented in [7, 12, 16], except that we use the finite element method instead of the finite difference method, as this more readily allows the use of adaptive refinement.

3. Discretization. To solve Equation (2.1), we use the following variational formulation, similar to that given in [6]: Find $u(t) \in V = H_0^1(\Omega)$, $t \in I$, such that

$$(\dot{u}(t), v)_\Omega + a(D(u(t)); u(t), v)_\Omega = < g, v >_{\partial\Omega} \qquad \forall v \in V, \tag{3.1}$$

where

$$a(D(u); v, w)_\Omega = \int_\Omega D(u) \nabla v \nabla w \, dx + \int_{\Gamma_{F_0}} \frac{vw}{2} \, dx + \int_{\Gamma_{F_1}} \frac{vw}{2} \, dx,$$

$$< v, w >_{\partial\Omega} = \int_{\partial\Omega} vw \, dx$$

and

$$g = \begin{cases} 0 & \text{on } \Gamma_N \times I, \\ 2 & \text{on } \Gamma_{F_0} \times I, \\ 0 & \text{on } \Gamma_{F_1} \times I. \end{cases}$$

The time derivative is dealt with through the use of either an implicit Euler or the Crank-Nicolson method.

4. Node-Edge Data Structure. A node-edge data structure similar to the one described in [15, 17, 18, 20] is used to store the finite element grid. In this data structure, a grid $\mathcal{M}^m$ is defined in terms of its geometrical, topological and algebraic attributes. The geometrical and topological attributes are simply the set of *nodes*, $\mathcal{N}^m$, and *edges*, $\mathcal{E}^m$. The stiffness matrix is associated with the algebraic attribute and is stored in the set of connections, $\mathcal{C}_A^m$, as a graph.

The node-edge data structure does not *explicitly* contain any information about the elements in the grid. Consequently, the same data structure can be used to store triangular, quadrilateral or tetrahedral grids. Information about the elements may be extracted by looping through the nodes and edges if necessary. The advantage of such a data structure is its flexibility. Although we concentrate on serial results in this paper, the code has been implemented in parallel and those results will be presented in a future paper.

We use a refinement algorithm to build the sequence of coarse grids needed by our solution techniques (nonlinear multigrid methods). That is, the new, refined grid becomes the fine grid in the multigrid algorithm and the old grid(s) is kept as the coarse grid(s). In terms of the notation defined above, this nested sequence of grids is represented by $\mathcal{M}^1 \subset \mathcal{M}^2 \subset ... \subset \mathcal{M}^n$. A more thorough description of this refinement algorithm is given in Section 6.

Information is moved from the coarse grid $\mathcal{M}^{m-1}$ to the fine grid $\mathcal{M}^m$ by the linear interpolation matrix $\mathbf{I}_{m-1}^m$. The restriction matrix $\mathbf{I}_m^{m-1}$ is defined to be the transpose of the interpolation matrix and moves the information from the fine grid $\mathcal{M}^m$ to the coarse grid $\mathcal{M}^{m-1}$. This extra algebraic information is stored in the table of connections. That is, if $\mathcal{C}_I^m$, $\mathcal{C}_R^m$ and $\mathcal{C}_A^m$ hold the interpolation, restriction and stiffness matrices, respectively, then the algebraic information for a multigrid grid is the set of *connections*, $\mathcal{C}^m$, where

$$\mathcal{C}^m = \mathcal{C}_A^m \cup \mathcal{C}_I^m \cup \mathcal{C}_R^m,$$

for $1 < m < n$, $\mathcal{C}^1 = \mathcal{C}_A^1 \cup \mathcal{C}_I^1$ and $\mathcal{C}^n = \mathcal{C}_A^n \cup \mathcal{C}_R^n$.

Finally, the grid $\mathcal{M}^m$ is given by $\mathcal{M}^m = \{\mathcal{N}^m, \mathcal{E}^m, \mathcal{C}^m\}$.

5. Solution Techniques. We compare two different solution techniques: the inexact Newton-multigrid method and the Full Approximation Storage multigrid (FAS). Note that Newton's method relies on a global linearization sweep whereas the FAS scheme uses local linearizations.

5.1. Inexact Newton-Multigrid. Our implementation is a standard implementation of Newton's method, but we have included a brief description below to aid our discussion of the numerical results.

Suppose we want to solve the nonlinear system $\mathbf{N}[\mathbf{x}] = \mathbf{b}$ where $\mathbf{N}$ is a nonlinear operator. Let $\mathbf{F}[\mathbf{x}] = \mathbf{b} - \mathbf{N}[\mathbf{x}]$ and take an initial guess $\mathbf{x}_0$. A high level algorithm for the inexact Newton-multigrid method is;

While $|\mathbf{F}[\mathbf{x}_k]| >$ a given tolerance

Calculate the Jacobian $\mathbf{F}'(\mathbf{x}_k)$

Move the Jacobian down the grid levels

Solve the linear system $\mathbf{F}'(\mathbf{x}_k)\mathbf{y} = -\mathbf{F}[\mathbf{x}_k]$ by using the multigrid method

Find the scaling factor γ
Set $\mathbf{x}_{k+1} = \gamma\mathbf{y} + \mathbf{x}_k$

The method is inexact because the linear system is not solved exactly; we just require that

$$||\mathbf{F}[\mathbf{x}_k] + \mathbf{F}'(\mathbf{x}_k)\mathbf{y}|| < ||\mathbf{F}[\mathbf{x}_k]||/10.$$

The scaling factor is defined as

$$\gamma = \min\left\{1, 1/||\mathbf{c}||_2\right\}^{\frac{1}{4}},$$

where $\mathbf{c}_i = \mathbf{y}_i/(\mathbf{x}_k)_i$. It is necessary to include a scaling factor, particularly when solving the time-dependent problems, as the solution changes rapidly near wave fronts. This is similar to the scaling factor defined in [16].

When using the multigrid algorithm to solve a linear system $\mathbf{A}^n\mathbf{y} = \mathbf{f}^n$ defined on a fine grid, we need to 'move' the system down to the coarse grids. One way of doing this is to set

$$\mathbf{A}^{m-1} = \mathbf{I}_m^{m-1}\mathbf{A}^m\mathbf{I}_{m-1}^m, \tag{5.1}$$

where $\mathbf{A}^m$ $(\mathbf{1} < m \leq n)$ is the matrix defined on the grid $\mathcal{M}^m$, and $\mathbf{I}_m^{m-1}$ and $\mathbf{I}_{m-1}^m$ are the interpolation and restriction matrices introduced in Section 4. The 'Move the Jacobian down the grid levels' line in the above algorithm means that we define the coarse grid matrices in this way.

5.2. FAS Scheme. A high level algorithm of the FAS scheme is given below. A more thorough description can be found in, for example, [1, 2, 8].

While $|\mathbf{F}[\mathbf{x}_k]|| >$ a given tolerance
  Apply a nonlinear smoother μ_1 times to the system $\mathbf{N}^m[\mathbf{x}^m] = \mathbf{b}^m$
  If not at coarsest grid
    Compute the defect $\mathbf{d}^m = \mathbf{b}^m - \mathbf{N}^m[\mathbf{x}^m]$
    Restrict the defect $\mathbf{d}^{m-1} = \mathbf{I}_m^{m-1}\mathbf{d}^m$
    Restrict the current approximation $\mathbf{x}^{m-1} = \widehat{\mathbf{I}}_m^{m-1}\mathbf{x}^m$
    Compute the approximation to
      $\mathbf{N}^{m-1}[\mathbf{y}^{m-1}] = \mathbf{d}^{m-1} + \mathbf{N}^{m-1}[\mathbf{x}^{m-1}]$
      by calling the FAS Scheme again using $\mathbf{x}^{m-1}$ as an initial guess
    Calculate the correction $\widehat{\mathbf{x}}^{m-1} = \mathbf{y}^{m-1} - \mathbf{x}^{m-1}$
    Interpolate the correction $\widehat{\mathbf{x}}^m = \mathbf{I}_{m-1}^m\widehat{\mathbf{x}}_m^{m-1}$
    Correct the current approximation $\mathbf{x}^m = \mathbf{x}^m + \widehat{\mathbf{x}}^m$
  Apply a nonlinear smoother μ_2 times to the system $\mathbf{N}^m[\mathbf{x}^m] = \mathbf{b}^m$

Notice that the Jacobian matrix is not formed, so the FAS scheme requires less storage than Newton's method.

The matrix $\widehat{\mathbf{I}}_m^{m-1}$ is a restriction operator, but not necessarily the same as $\mathbf{I}_m^{m-1}$. In this work we used injection for the operation.

The results presented in this paper were obtained by using a nonlinear SOR method [13] as a smoother. The linearization phase is incorporated in the smoother by applying a point-Newton method to a given grid node after 'fixing' the value at all of the other nodes. To apply the point Newton method, the diffusivity term (and its derivative) must be evaluated, which is expensive.

6. Grid Refinement. The refinement algorithm is based on the newest node bisection method. In this method, the triangles are subdivided by bisecting the edges that sit opposite the newest nodes. For example, if the dark points in Figure 6.1 are the newest nodes, then the resulting grid after one and two refinement sweeps are shown in Figure 6.2.

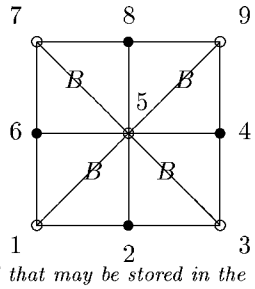


FIG. 6.1. *Example grid that may be stored in the node-edge data structure.*

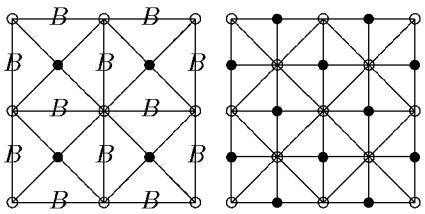


FIG. 6.2. *Resulting grid after two non-adaptive refinement sweeps of the grid in Figure 6.1. Note that the edges have been bisected along the base edges marked by a B.*

In terms of the node-edge data structure, it is easier to work with the base edges rather then the newest nodes, where the base edges are the edges that sit opposite the newest nodes, such as those marked by B in Figure 6.1.

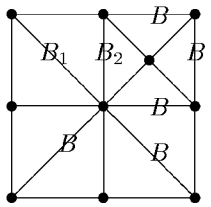


FIG. 6.3. *Result after bisecting the grid in Figure 6.1 along one of the base edges.*

6.1. Controlling the Order of Refinement. The most difficult part of the adaptive refinement routine is ensuring that the edges are bisected in the correct order to avoid long thin triangles. For example, suppose a triangle in Figure 6.1 is refined along one of the base edges as shown in Figure 6.3; then several of the triangles in the resulting grid will have two base edges. If the edges B_1 and B_2 are bisected during the next refinement sweep, then it is not clear which base edge should be bisected first. If the wrong edge is chosen, as in Figure 6.4, we get long thin triangles which reduce the efficiency of the grid.

To determine the order in which to bisect the edges, we use a method similar to Mitchell's Compatibly Divisible Triangles [9, 10, 11].

We define an interface-base edge to be an edge that sits between two different levels of refinement. For example, in Figure 6.5, we have redrawn the grid from Figure 6.3 and marked the interface-base edges by

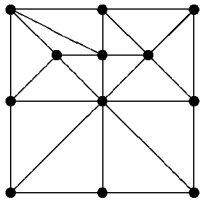


FIG. 6.4. *If the wrong edge (B_1) is bisected first, then the triangles can become long and thin.*

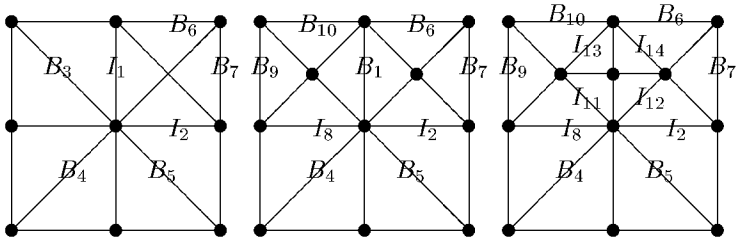


FIG. 6.5. *The interface-base edges marked by I sit between two different levels of refinement. Once an interface-base edge has been updated to a base edge, it may be bisected.*

an I. The neighboring coarse triangles must be refined before the interface-base edges are bisected. So B_3 in Figure 6.5 must be bisected before I_1. Note that it may be necessary to refine more then one neighboring triangle. For example, to bisect edge I_{11} in Figure 6.5, edges B_4 and I_8 should be bisected first.

6.2. Error Indicator (Stationary Problem). The idea behind our error indicator is to determine if the addition of a new node will significantly reduce the error.

Let $\mathbf{v}^m$ be the current approximation to the system of equations $\mathbf{N}^m[\mathbf{u}^m] = \mathbf{b}^m$, where $\mathbf{N}^m$ and $\mathbf{b}^m$ are the stiffness matrix and load vector defined on the current grid, $\mathcal{M}^m$. Then, each base edge and interface-base edge is assigned an error indicator which is equal to a weighted residual calculated at its midpoint. That is, if node i is the midpoint of an edge then the error indicator e^m assigned to that edge is:

$$e^m = \frac{\mathbf{r}_i^{m+1}}{\mathbf{N}_{i,i}^{m+1}[\mathbf{I}_m^{m+1}\mathbf{v}^m]}, \tag{6.1}$$

where

$$\mathbf{r}^{m+1} = \mathbf{b}^{m+1} - \mathbf{N}^{m+1}[\mathbf{I}_m^{m+1}\mathbf{v}^m].$$

$\mathbf{N}^{m+1}$ and $\mathbf{b}^{m+1}$ are the resulting stiffness matrix and load vector that would result if the edge was bisected at node i to form a new set of triangles. Note that it is not necessary to construct the whole stiffness matrix (or load vector); we only need the row corresponding to node i.

The motivation behind the error indicator is the question: how much will the error be reduced if we add node i? In regions of the domain where the solution is well approximated by the coarse grid we would expect the residual $\mathbf{r}^{m+1}$ to be small; in other regions where the solution is rapidly changing, and not well approximated by the coarse grid, the residual will be large. We divide by $\mathbf{N}_{i,i}^{m+1}[\mathbf{I}_m^{m+1}\mathbf{v}^m]$ to normalize the residual.

This error indicator is similar to the error indicator described by Mitchell [9, 10, 11], Rüde [15] or Villegas [21].

6.3. Moving Grids. When modeling non-stationary problems it is advantageous to adjust the shape of the grid to match the movement of the wave front.

One approach to building such grids is to first de-refine the grid and then re-refine it to take the movement of the wave front into account. By noting that we store a sequence of nested grids, not just a single FEM grid, we are able to implement a very cheap de-refinement procedure. We simply shuffle the grid up one level, i.e. set $\bar{\mathcal{M}}^m$ to $\mathcal{M}^{m-1}$ where $\bar{\mathcal{M}}^m$ is the de-refined grid. The interpolation and restriction information at the coarsest and finest grids has to be updated, but otherwise this is a simple copy. Once all of the levels have been updated, we can apply the refinement algorithm described above.

6.4. Error Indicator (Non-Stationary Problem). The next issue is how to calculate the error indicator when taking the time derivative into account. To calculate the indicator at the next time-step, we need an approximation of the solution at that time-step. Applying an implicit algorithm to approximate the solution is cost-prohibitive, so we use an explicit method instead (we only use an explicit method to find the error indicator; once the grid has been refined, we use an implicit method to calculate the solution). Recall that the explicit Euler method is

$$\mathbf{M}^m \bar{\mathbf{v}}^m = \mathbf{M}^m \mathbf{v}^m + \Delta t(\mathbf{b}^m - \mathbf{N}^m[\mathbf{v}^m]),$$

where $\mathbf{M}^m$ is the mass matrix, $\mathbf{v}^m$ is the solution at the current time-step, $\bar{\mathbf{v}}^m$ is the solution at the next time-step and Δt is the step size. Based on this equation we then define the error indicator for non-stationary problems to be:

$$\begin{aligned} e^m &= \frac{\mathbf{r}_i^{m+1}}{\left(\mathbf{M}_{i,i}^{m+1}\right)^{1/4}}, \\ \mathbf{r}^{m+1} &= \mathbf{g}^{m+1} - \mathbf{M}^{m+1}\mathbf{I}_m^{m+1}\bar{\mathbf{v}}^m, \\ \mathbf{g}^{m+1} &= \mathbf{I}_m^{m+1}\left(\mathbf{M}^m\mathbf{v}^m + \Delta t(\mathbf{b}^m - \mathbf{N}^m[\mathbf{v}^m])\right) \end{aligned} \tag{6.2}$$

Note that this indicator only needs to evaluate $\mathbf{N}$ once on the original de-refined grid, to form the right-hand-side, and is thus a lot cheaper than the indicator used for the stationary problem.

Figures 6.6, 6.7 and 6.8 show examples of the moving grid. To obtain these examples we set $\alpha = -1$, $\beta = 1/4$. The values of Z are as given in Figure 2.1 and $D(E) = D_1(E)$. The step size is $0.5/32$, and the figures show the results at time step 10, 20 and 30 respectively. There is an increase in the number of nodes over time, however this is consistent with the shape of the solution. The indicator should pick up the regions where the solution is changing rapidly, which it does.

7. Results. We ran all of the test problems on a Beowulf cluster consisting of 400 MHz Pentium II processors with 384 MB of 100 MHz RAM. There are 32 such CPU nodes available in the cluster. Further particulars of this machine can be found at `http://www.icase.edu/`.

7.1. Solution of Stationary Problem. To better understand the behavior of the two nonlinear multigrid methods we firstly considered the stationary problem.

7.1.1. Test Problems. We looked at four different test problems, `low`, `low J`, `high` and `high J`.

In the first two examples, `low` and `low J`, we set $\alpha = -1$ and $\beta = 1/4$. The difference between the two examples is in the definition of the atomic mass number Z. In `low` Z is fixed at 10 throughout the whole domain, but in `low J` the values for Z vary as shown in Figure 2.1. Figure 7.1 gives example solutions of `low` and `low J`.

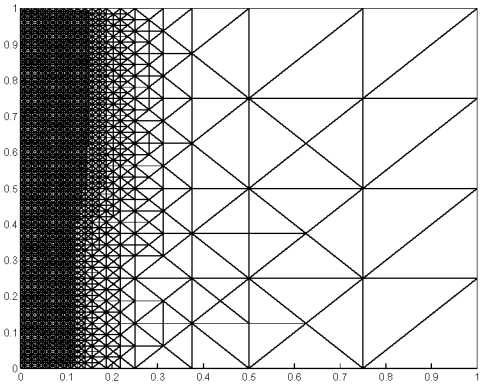


FIG. 6.6. *Example grid at time step 10, contains 2868 nodes.*

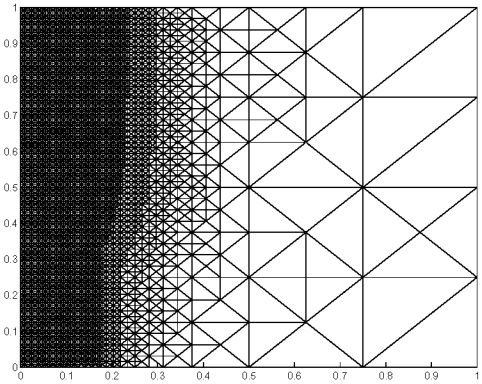


FIG. 6.7. *Example grid at time step 20, contains 4390 nodes.*

The other two examples, `high` and `high J`, are much more difficult to solve. In this case the nonlinearity and jump size were increased to $\alpha = -3$ and $\beta = 3/4$. Once again, Z is fixed at 10 for `high`, but is spatially dependent for `high J`. Figure 7.2 shows example solutions of `high` and `high J`.

In all of the test problems the flux limiter was included in the definition of the diffusivity term as shown in Equation (2.3).

7.1.2. Newton's method. Let us firstly look at the results for Newton's method, which are given in Tables 7.1 and 7.2.

The timing results have been broken down into the total time, time required to solve the nonlinear system using Newton's method, the time spent solving the linearized system with the aid of the V-scheme and the time needed to build the nested sequence of grids.

Before adding a new grid level, $\mathcal{M}^m$, the problem was solved on the coarse grid $\mathcal{M}^{m-1}$ and the result was interpolated up as an initial guess on $\mathcal{M}^m$. The times given in the tables are the accumulated time over all of the grid levels (i.e. not just the time to solve the problem on the finest grid).

The number of iterations, labeled as 'No. It.', is the number of iterations required to solve the problem on the finest grid level. The iterations were terminated when the residual $\mathbf{F}[\mathbf{x}_k]$ was less than 10^{-5} or the number of iterations reached a maximum of 10. Typically the method did not converge within 10 iterations on the coarser grids, but we overlooked this as the coarse grids are only used to give an initial guess to the fine grid solution. The method did not converge on the finest grid for test problem `high J` when uniform grid refinement was used.

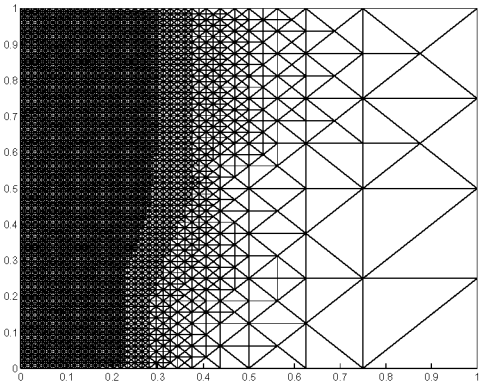


FIG. 6.8. *Example grid at time step 30, contains 5602 nodes.*

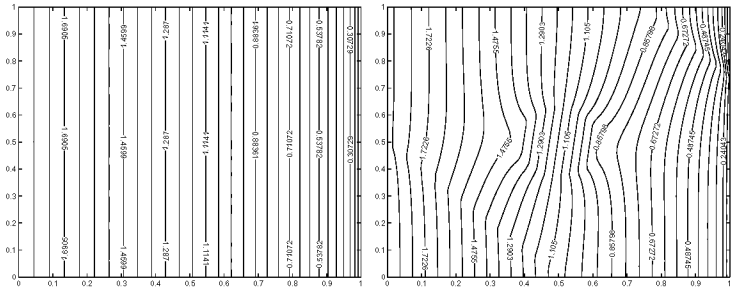


FIG. 7.1. *Example solutions of the test problems* `low` *and* `low J` *respectively.*

One of the difficulties in solving this particular problem is that Newton's method tries to 'push' the solution down below zero, but we can not pass negative energy values into the diffusivity functions, $D_1(E)$ or $D_2(E)$. To avoid the negative values we limit how small the energy may become, but this slows down or halts the convergence rate of Newton's method on the coarser grids.

Two pre-smoothers and two post-smoothers were used in the V-scheme solver. The convergence rate for the solver was close to 0.5 for all of the test problems.

Six levels of uniformly refined grids were used to obtain the results presented in Table 7.1. The number of nodes on the finest grid was 66049.

Five levels of adaptively refined grids form the basis of the results given in Table 7.2. The number of nodes on the finest grid level ranges from 17043 to 50785; which is an artifact of the refinement routine. Refinement continues on each grid level until the error indicator has been reduced by a factor of four, which does not necessarily imply that the number of nodes increases by a factor of four from one grid level to the next. The time required to build these grids is a lot higher then we would like, mainly due to the cost of calculating the error indicator, Equation (6.1). As discussed in the following section, calculations involving the diffusivity term $D(E)$ are expensive.

In Table 7.1 we see that Newton's method slows down as the strength of the nonlinearity is increased, as expected. The relatively small coefficient jumps seen in `low J` have little effect, but the method struggled as the jump size was increased.

The results displayed in Table 7.2 are a little more erratic. The adaptive refinement aided the solution method when there was a strong nonlinearity. The timings reported for `high` are low because the coarse grids were fine enough to capture the shape of the solution, and consequently the initial guess interpolated up to the fine grid was well within the ball of convergence.

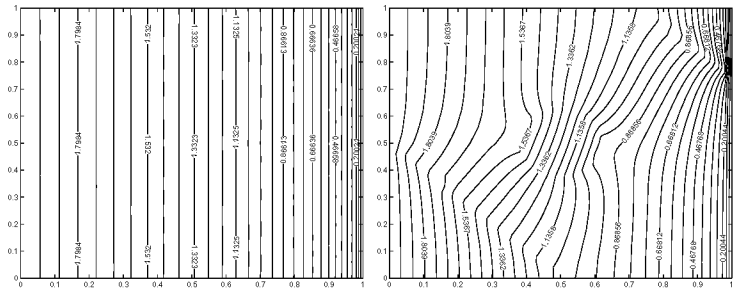


FIG. 7.2. *Example solutions of the test problems* `high` *and* `high J` *respectively.*

TABLE 7.1

Solution time, in seconds, when solving the test problems using Newton's method on an uniformly refined grid.

	low	low J	high	high J
No. Nodes	66049	66049	66049	66049
No. It.	3	3	6	*
Total Time	164.7	143.5	412.7	646.1
Newton	138.7	117.5	386.9	620.2
V-scheme	44.2	24.3	135.9	237.1
Grid Refine	12.2	12.2	12.1	12.2

7.1.3. FAS. Tables 7.3 and 7.4 give the timing results when the FAS scheme was used. Much of the discussion given in Section 7.1.2 on how the timing results were obtained also applies here.

The results labeled 'Nonlin. SOR' show the amount of time spent in the nonlinear SOR routine. Two pre-smoothers and two post-smoothers were used for `low` and `low J`, four pre-smoother and four post-smoothers were used for `high` and `high J`. A weight of 0.8 was set in the nonlinear SOR method. The maximum number of FAS iterations allowed on each level was 6.

The FAS Scheme did not converge for `high J` when either uniform or adaptive refinement was used. We have not investigated this further yet because the non-stationary problem is our main focus.

The times for the FAS scheme are slower than Newton's method, except for `high` with uniform refinement. This does not mean that the FAS scheme 'performed poorly' in fact it had a convergence rate of 0.5, similar to the linear solve in Newton's method. We also measured the number of times each node was updated during the pre-smoothing and post-smoothing stages, for uniformly refined grids, and found that the FAS scheme had fewer or equal number of updates compared to Newton's method. The main reason why FAS is slower is because of the high cost of calculating the effect of the diffusivity term, and its derivative. Every time the value of a given node is changed, $a(D(u); v, w)$ has to be reevaluated; which involves a search through the data structure to find the appropriate triangles, the formulation of the basis functions and the evaluation of the integral. In Newton's method we only have to evaluate the term once for each node on the finest grid level to form the Jacobian, but with the FAS scheme it is necessary to evaluate the term several times for each node on each grid level.

7.2. Non-Stationary. The Crank-Nicolson method was used in all of the test runs.

7.2.1. Uniform v's Adaptive. To determine if the use of adaptive refinement influences the behavior of the solution we looked at the solution obtained on a uniformly refined grid and compared it with one obtained on an adaptively refined grid. The flux limiter was included in the diffusivity term and $\alpha = -3$

TABLE 7.2
Solution time, in seconds, when solving the test problems using Newton's method on an adaptively refined grid.

	low	low J	high	high J
No. Nodes	41682	24737	50785	17043
No. It.	3	3	1	8
Total Time	156.8	106.3	74.2	125.2
Newton	105.4	68.2	17.4	105.1
V-scheme	36.9	24.5	0.93	24.5
Grid Refine	42.2	32.6	45.7	16.6

TABLE 7.3
Solution time, in seconds, when solving the test problems using the FAS scheme on an uniformly refined grid.

	low	low J	high	high J
No. Nodes	66049	66049	66049	66049
No. It.	2	2	1	*
Total Time	310.6	318.6	318.4	*
FAS	284.7	292.3	292.7	*
Nonlin. SOR	222.1	228.0	252.3	*
Grid Refine	12.1	12.1	12.1	*

and $\beta = 3/4$. Figures 7.3 and 7.4 show the results at $t = 0.05$ with time steps $\Delta t = 0.1/32$. The atomic mass number Z was fixed at 10, meaning that the solution is homogeneous along the y-axis (see for example the solution of `low` and `high` given in Figures 7.1 and 7.2). The graphs shown in Figures 7.3 and 7.4 are a slice taken along the $y = 0$ plane.

As stated in [16], when the flux limiter is included the problem changes from being locally parabolic to locally hyperbolic and thus becomes more difficult to solve. This property is reflected in the graph shown in Figure 7.3 where we see a small oscillation near the left boundary. By using adaptive refinement we are able to decrease the grid size near that boundary and thus remove the oscillation.

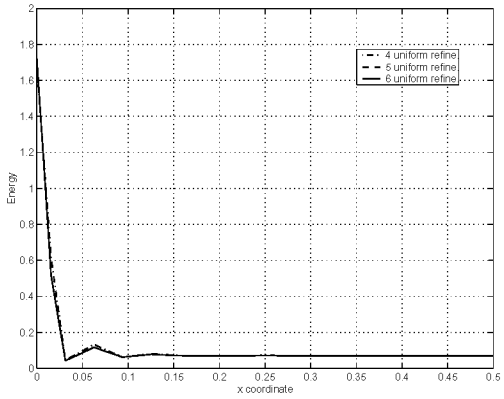


FIG. 7.3. *Example solutions of the test problems* `low` *and* `low J` *respectively.*

7.2.2. Jump Size and Non-Stationary Solutions. We now look at how the jump size affects the movement of the energy wave. We used the inexact Newton-multigrid method to solve the problem.

TABLE 7.4
Solution time, in seconds, when solving the test problems using FAS scheme on an adaptively refined grid

	low	low J	high	high J
No. Nodes	32729	25690	50682	*
No. It.	1	3	1	*
Total Time	150.3	234.5	218.2	*
FAS	106.9	196.2	161.9	*
Nonlin. SOR	82.9	151.9	138.1	*
Grid Refine	36.0	32.4	45.6	*

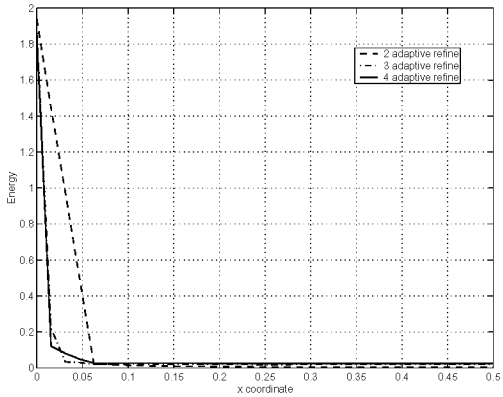


FIG. 7.4. *Example solutions of the test problems* `low` *and* `low J` *respectively.*

In the first example $\alpha = -2$ and $\beta = 3/4$. The values of Z are as given in Figure 2.1 and a flux limiter was used. We took time steps of size 3/64 and the results in Figures 7.5 and 7.6 show the energy at time steps 32 and 64 respectively. The nonlinear iterations were terminated when the residual was less than 10^{-6}. The examples took 13202 seconds to run.

Recall that the atomic mass number in the lower left corner of the domain is higher than that in the upper left corner. We can clearly see how this influences the movement of the wave front.

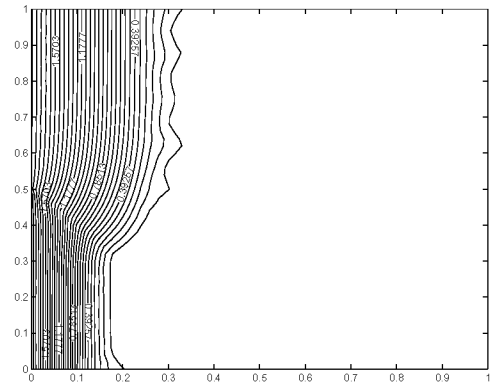


FIG. 7.5. *Example solution at time step 32 with* $\alpha = -2$.

In the next set of examples we set $\alpha = -3$ and $\beta = 3/4$. All of the other parameters are the same as those given above. The results in Figures 7.7 and 7.8 show the energy at time steps 32 and 64 respectively. It took a total of 3520 seconds to solve the problem, with approximately 89% of that time spent in the

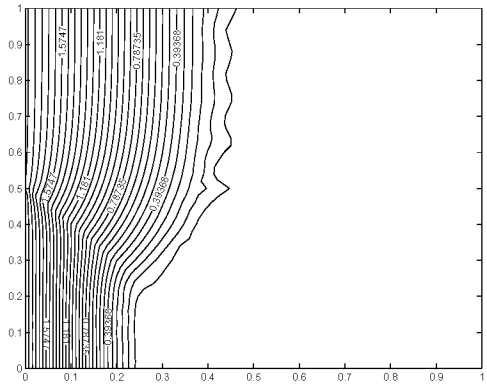


FIG. 7.6. *Example solution at time step 64 with* $\alpha = -2$.

nonlinear solver and 7% spent in the adaptive refinement routine. The number of nodes on the finest grid level varied from about 2500 to 5500.

These graphs show how an increase in the mass number slows down the movement of the wave front.

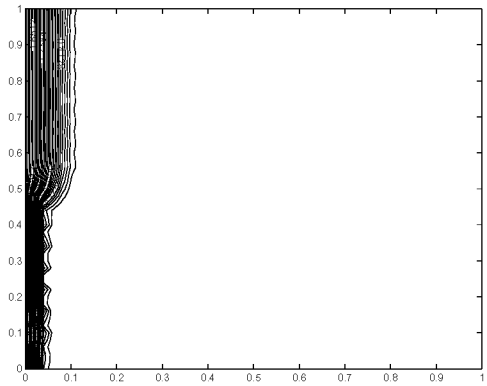


FIG. 7.7. *Example solution at time step 32 with* $\alpha = -3$.

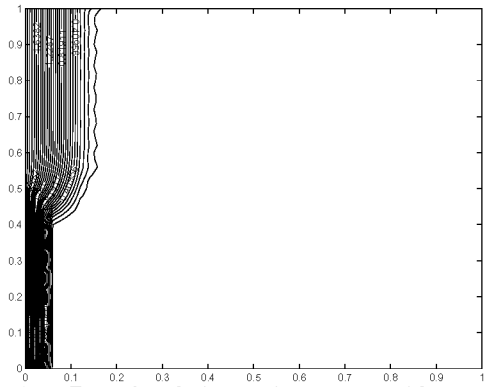


FIG. 7.8. *Example solution at time step 64 with* $\alpha = -3$.

8. Conclusion. The inexact Newton-multigrid method is faster than the FAS scheme when solving radiation transport equations. Note however, we only presented serial results in this report and our initial study into the parallel performance suggests that the FAS scheme scales better.

Adaptive refinement helps to capture the information along the wave front.

9. Future Work. Both of the solvers have difficulty converging during the first few time steps, thus we like to investigate some adaptive time-stepping techniques.

We would also like to study the parallel scalability of the solvers.

REFERENCES

[1] A. Brandt, Multi-level adaptive solutions to boundary-value problems. *Math Comput.*, 31 (1977), pp. 333–390.

[2] W. L. Briggs, V. E. Henson, and S. F. McCormick, *A Multigrid Tutorial.* SIAM, 2000.

[3] P N. Brown, B. Chang, F. Graziani and C. S. Woodward, *Implicit solution of large-scale radiation-material energy transfer problems*, Technical Report UCRL-JC-132831, Lawrence Livermore National Laboratory, Jan. 1999. To appear in Proceedings of the Fourth IMACS International Symposium on Iterative Methods in Scientific Computations.

[4] P N. Brown and C. S. Woodward, *Preconditioning Strategies for Fully Implicit Radiation Diffusion with Material-Energy Transfer*, Technical Report UCRL-JC-139087, Lawrence Livermore National Laboratory, May. 2000. To appear in SIAM J. Sci. Comput.

[5] D. A. Knoll, W. J. Rider and G. L. Olson, *An efficient nonlinear solution method for non-equilibrium radiation diffusion*, Technical Report LA-UR-98-2154, Los Alamos National Laboratory, 1998. Submitted to J. Quant. Spec. and Rad. Trans.

[6] M. Křížek and L. Liu, *Finite element approximation of a nonlinear heat conduction problem in anisotropic media*, Technical Report 4/1997, Laboratory of Scientific Computing, University of Jyväskylä, Finland, 1997.

[7] D. J. Mavriplis, *Multigrid approaches to non-linear diffusion problems on unstructured meshes*, ICASE Report No. 2001-3, (NASA/CR-2001-210660), February 12, 2001, 16 pages. Submitted to the Journal of Numerical Linear Algebra with Applications.

[8] S. F. McCormick, *Multigrid Methods.* SIAM Frontiers In Applied Mathematics, 1987.

[9] W. F. Mitchell, *Unified Multilevel Adaptive Finite Element Methods For Elliptic Problems.* PhD thesis, Department Of Computer Science, University Of Illinois at Urbana-Champaign, Urbana, IL, 1988. Technical Report UIUCDCS-R-88-1436.

[10] W. F. Mitchell, A comparison of adaptive refinement techniques for elliptic problems. *ACM Trans. Math. Software*, 15 (1989), pp. 326–347.

[11] W. F. Mitchell, Optimal multilevel iterative methods for adaptive grids. *SIAM J. Sci. Stat. Comput*, 13 (1992), pp. 146–167.

[12] V. A. Mousseau, D. A. Knoll and W. J. Rider, Physics-based preconditioning and the Newton-Krylov method for non-equilibrium radiation diffusion. *Journal of Computational Physics*, 160 (2000), pp. 743-765.

[13] J. M. Ortega and W. C. Rheinboldt, *Iterative solution of nonlinear equations in several variables*, Academic Press, New York and London, 1970.

[14] G. C. Pomraning, *The equations of radiation hydrodynamics*, Pergamon, New York, 1973.

[15] U. Rüde, *Mathematical and computational techniques for multilevel adaptive methods*, SIAM, Philadelphia, 1993.

[16] W. J. Rider, D. A. Knoll and G. L. Olson, *A multigrid Newton-Krylov method for multimaterial equilibrium radiation diffusion*, Technical Report LA-UR-98-2153, Los Alamos National Laboratory,

1998, 34 pages.

[17] L. STALS, *Adaptive multigrid in parallel*, in Proceedings of Seventh SIAM Conference on Parallel Processing for Scientific Computing, D. Bailey, P. Bjørstad, J. Gilbert, M. Mascagni, R. Schreiber, H. Simon, V. Torczon and L. Watson, eds, SIAM, Philadelphia, 1995, pp. 367-372.

[18] L. STALS, *Parallel multigrid on unstructured grids using adaptive finite element methods*, PhD thesis, Department of Mathematics, Australian National University, Canberra, Australia, 1995.

[19] L. STALS, *Implementation of multigrid on parallel machines using adaptive finite element methods*, in Proceedings of 9th International Conference on Domain Decomposition, P. Bjørstad, M. Espedal and D. Keyes, eds, 1998, pp. 488-496.

[20] L. STALS, *A flexible data structure for the adaptive refinement of unstructured grids in parallel*, in The Proceedings of The 14th Kiel GAMM Seminar on Concepts of Numerical Software, January, 1997. To appear.

[21] J. C. A. VILLEGAS, *Anisotropic Adaptive Refinement Algorithms for Finite Element Methods*. PhD thesis, Graduate School of Arts and Science, Department of Mathematics, New York University, September 2000.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*

2. REPORT DATE
October 2001

3. REPORT TYPE AND DATES COVERED
Contractor Report

4. TITLE AND SUBTITLE
The solution of radiation transport equations with adaptive finite elements

5. FUNDING NUMBERS
C NAS1-97046
WU 505-90-52-01

6. AUTHOR(S)
Linda Stals

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
ICASE
Mail Stop 132C
NASA Langley Research Center
Hampton, VA 23681-2199

8. PERFORMING ORGANIZATION REPORT NUMBER
ICASE Report No. 2001-29

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
National Aeronautics and Space Administration
Langley Research Center
Hampton, VA 23681-2199

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
NASA/CR-2001-211230
ICASE Report No. 2001-29

11. SUPPLEMENTARY NOTES
Langley Technical Monitor: Dennis M. Bushnell
Final Report
Submitted to the Proceedings of the Tenth Copper Mountain Conference on Multigrid Methods.

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Unclassified–Unlimited
Subject Category 60, 61
Distribution: Nonstandard
Availability: NASA-CASI (301) 621-0390

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*
We compare the performance of an inexact Newton-multigrid method and Full Approximation Storage multigrid when solving radiation transport equations. We also present an adaptive refinement algorithm and explore its impact on the solution of such equations.

14. SUBJECT TERMS
FAS, multigrid, Newton method, radiation transport

15. NUMBER OF PAGES
20

16. PRICE CODE
A03

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified		